**Amendments to the Specification:**

(A) Please replace the paragraph that starts on page 1, line 20, with the following amended paragraph:

The present ~~invention~~ disclosure relates to a data structure and method for debugging software systems and, in particular to a data structure and method that represents the static aspects of a software system and identifies control constraint conflicts in the software system.

(B) Please replace the heading on page 1, line 24, with the following amended heading:

BACKGROUND ~~OF THE INVENTION~~

(C) Please replace the heading on page 7, line 24, with the following amended heading:

SUMMARY ~~OF THE INVENTION~~

(D) Please replace the paragraph that starts on page 8, line 3, with the following amended paragraph:

The present ~~invention provides~~ disclosure describes a data structure and method for representing the static aspects of a software system and identifying control constraint conflicts in the system. The techniques presented here can be implemented in the PIA tools and enabled by the methodology introduced above. Each technique can eliminate bugs prior to execution.

(E) Please replace the four paragraphs from page 13, line 22, through page 14, line 2, with the following four amended paragraphs:

FIG. 44A is an edge that asserts the value ~~true~~ false at its head and is responsive to the value true at its tail.

FIG. 44B is an edge that asserts the value ~~true~~ false at its head and is responsive to the value false at its tail.

FIG. 44C is an edge that asserts the value ~~false~~ true at its head and is responsive to the value true at its tail.

FIG. 44D is an edge that asserts the value ~~false~~ true at its head and is responsive to the value false at its tail.

(F) Please replace the paragraph that starts on page 86, line 3, with the following amended paragraph:

A static control graph (SCG) is a graph-theoretic representation of all pure control constraints. FIG. 43 shows a simple SCG. It is a bi-partite digraph, having two types of nodes: conjunctive nodes ~~4300~~ 4316, 4330, and 4332, which, as the name implies, produce results only when all incident edges ~~4302~~ 4310 are satisfied, and disjunctive nodes 4300, 4302, 4304, 4306, and 4308, which produce results if any incident edge ~~4302~~ 4310 is satisfied. Disjunctive nodes 4300, 4302, 4304, 4306, and 4308 correspond to modes 102 in components 100 and coordinators 410 (as previously shown in FIG. 1 and FIG. 4) throughout the system. An SCG for a complete system simultaneously represents all control constraints.

(G) Please replace the five paragraphs on page 86, line 13, through page 87, line 8, with the following five amended paragraphs:

- C is a set of conjunctive nodes ~~4300~~.
- D is a set of disjunctive nodes ~~4304, 4306, and 4308~~.

- ~~$E \subseteq [\{T_f, T_t\} x \{H_f; Ht\} x ((C x D) \cup (D x C))]$~~ $E \subset [ \{T_f, T_t\} \text{ x } \{H_f, H_t\} \text{ x } ((C \text{ x } D) \cup (D \text{ x } C)) ]$ is a set of directed, labeled edges ~~4302~~. ~~Edge~~ Edges are sensitive to either a false value or a true value at their tail ~~4310~~ 4311 ($T_f$ or $T_t$) and enforce either a false value or a true ~~true~~ value at their head 4312 ~~($H_f$ or $H_t$)~~ ($H_f$ or $H_t$). These are represented visually by a bubble at the appropriate end for a false value or the lack of a bubble for a true value.

An edge ~~4302~~ 4310 in an SCG can be either enabled or disabled; it produces the value true or the value false. Fig.44 illustrates a graphic notation for edge labels. Edge 4400 and 4402 marked with a bubble on head 4312, as in FIG. 44A and FIG. 44B, assert the value false when activated. When there is no mark on head 4312, as in FIG. 44C and FIG. 44D, an edge 4404 and 4406 asserts the value true when activated. A bubble on tail ~~4310~~ 4311, as in FIG. 44B and FIG. 44D, indicates that edge 4406 and 4402 is sensitive to false on the node it exits. The lack of such bubbles, as in FIG. 44A and FIG. 44C, indicates that 4404 and 4400 is sensitive to true.

Referring back to FIG. 43, the figure shows a simple SCG in which a node d 4306 must be active whenever a conjunction (a b c) ~~4314~~ 4332 is active and inactive whenever ~~anode~~ a node e 4308 is active. Although this looks similar to a Boolean network, it differs because the SCG edged represent implication, not connection. This is illustrated in FIG. 45, which shows a Boolean network OR node 4500; when all inputs and outputs are negated, it is equivalent to an AND node 4502 (by DeMorgan's). A disjunctive SCG node with all inputs and outputs negated 4504 is equivalent to a disjunctive node with no inputs and outputs negated 4506.

(H) Please replace the paragraph starting on page 89, line 12, with the following amended paragraph:

SCGs are similar to Petri-nets ~~[80]~~ in several significant ways. ~~petri-nets~~ Petri-nets are also bi-partite digraphs with the node types-transitions and places. A system state is represented by a marking of places, where each place can be either marked by a token or not marked at all. A transition fires if there is a token on each of the places on the opposite side of incoming edges. When a transition fires, it consumes all enabling ~~tokens and~~ tokens, and places tokens on each of the places on the opposite side of outgoing edges.